УДК 004.272.2:519.63

О.А. Дмитриева
Донецкий национальный технический университет
МОНМС Украины, г. Донецк
Украина, 83000, г. Донецк, ул. Артема, 58

Высокоэффективные алгоритмы управления шагом на основе параллельных коллокационных блочных методов

O.A. Dmitrieva
Donetsk National Technical University
MESYS of Ukraine and MAS of Ukraine, c. Donetsk
Ukraine, 83000, c. Donetsk, Artema st., 58

High-Performance Algorithms of Step Control Based on the Parallel Collocation Block Methods

О.А. Дмитрієва
Донецький національний технічний університет
МОНМС України, м. Донецьк
Україна, 83000, м. Донецьк, вул. Артема, 58

Високоефективні алгоритми управління кроком на основі паралельних колокаційних блокових методів

В статье предлагаются новые коллокационные блочные методы решения жестких систем обыкновенных дифференциальных уравнений с автоматическим управлением размером шага, которые изначально ориентированы на параллельную архитектуру. Основная идея, на которой базируется конструирование предлагаемых методов, заключается в одновременном получении приближений точного решения во всех расчетных точках блока.
Ключевые слова: задача Коши, параллельные вычисления, точки коллокации, порядок аппроксимации.

In this paper, we propose the new block collocation methods for solving stiff systems of ordinary differential equations with automatic step size, which initially are focused on the parallel architecture. The basic idea, which underlies the development of the proposed methods, is to obtain simultaneously the approximation of the exact solution in all the estimated points of the block.
Key Words: Cauchy problem, parallel computing, collocation points, the order of approximation.

У статті пропонуються нові колокаційні блокові методи розв’язання жорстких систем звичайних диференціальних рівнянь із автоматичним управлінням розміром кроку, які орієнтовані на паралельну архітектуру. Основна ідея, на якій базується конструювання пропонованих методів, полягає в одночасному одержанні наближень точного розв’язку у всіх розрахункових точках блоку.
Ключові слова: задача Коші, паралельні обчислення, точки колокації, порядок апроксимації.

Введение

Основные усилия при параллельном моделировании динамических объектов большой размерности, описываемых системами обыкновенных дифференциальных уравнений (СОДУ), в настоящее время концентрируются на распараллеливании последовательных алгоритмов и программ [1], [2].

При этом исследуются только информационные взаимодействия смежных частей алгоритма, и не уделяется должного внимания общей информационной структуре задачи, что не позволяет создать масштабируемые параллельные программы [3], [4].

Еще одной проблемой при моделировании динамических объектов, особенно тех, которые формализуются жесткими системами ОДУ, является необходимость изменения шага интегрирования [5], [6].

Также актуальными являются вопросы соотнесения погрешностей результатов и времени, затрачиваемого на получение решения.

Практически все известные в настоящее время численные методы с автоматическим выбором шага интегрирования основаны на вычислении главного члена локальной ошибки и последующем выборе такого размера для очередного шага, который является максимальным для заданного предела локальной точности [5-8].

Но при этом возникает необходимость повторных вычислений в каждой точке, что приводит к значительному увеличению вычислительных затрат.

Кроме того, жестко регламентируются пропорции сокращения шага [9], [10].

В статье предлагаются новые блочные методы решения систем обыкновенных дифференциальных уравнений, которые изначально ориентированы на параллельную архитектуру [11].

Основная идея, на которой базируется конструирование блочных методов для решения СОДУ на параллельных компьютерах, заключается в одновременном получении приближений точного решения в точках, образующих блок.

Поскольку точки внутри блока расположены регулярно, есть возможность определения и сопоставления локальных погрешностей во всех точках блока, в отличие от стадийных методов, где стадии, как правило, не совпадают, и сравнение ведется только по одной конечной расчетной точке t_{n+1}.

Также непривлекательным является использование в компьютерах с распределенной памятью мелкозернистого параллелизма стадийных методов.

Алгоритмы управления шагом, предлагаемые в статье, базируются на использовании коллокационных блочных одношаговых и многошаговых методов. Параллельный счет осуществляется в пределах одного цикла для всех точек блоков с размерностями s и $s+1$.

Две нити вычислений проходят независимо, а необходимость в обменах возникает только после получения конечных результатов для блоков расчетных точек.

Цель данной статьи состоит в создании коллокационных блочных расчетных схем, позволяющих обеспечивать автоматическое управление шагом интегрирования при моделировании динамических объектов в параллельных вычислительных системах.

Генерация коэффициентов расчетных схем коллокационных блочных методов

Для параллельной реализации численного решения задачи Коши

$$x' = f(t,(x(t)), \quad x(t_0) = x_0 \tag{1}$$

можно обобщить подходы, предложенные в [12], [13], и основанные на одношаговом и многошаговом блочных методах. При этом не обязательным является требование, связанное с совпадением размерностей опорного и расчетного блоков. Будем считать, что схема одношагового коллокационного блочного метода содержит s узлов в

рассчитываемом блоке (рис. 1) и одну опорную точку, многошагового – *s* узлов в рассчитываемом блоке при использовании вычисленных значений приближенного решения в *m* предшествующих блоку точках (рис. 2).

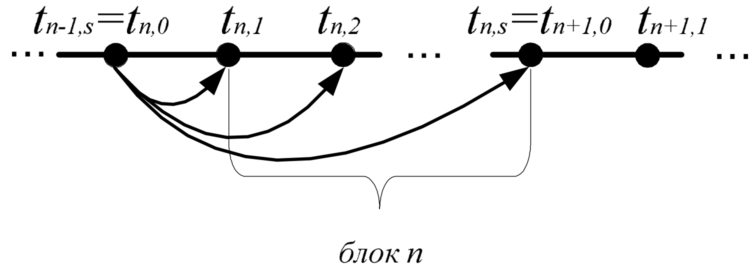


Рисунок 1 – Шаблон разностной схемы одношагового *s*-точечного блочного метода

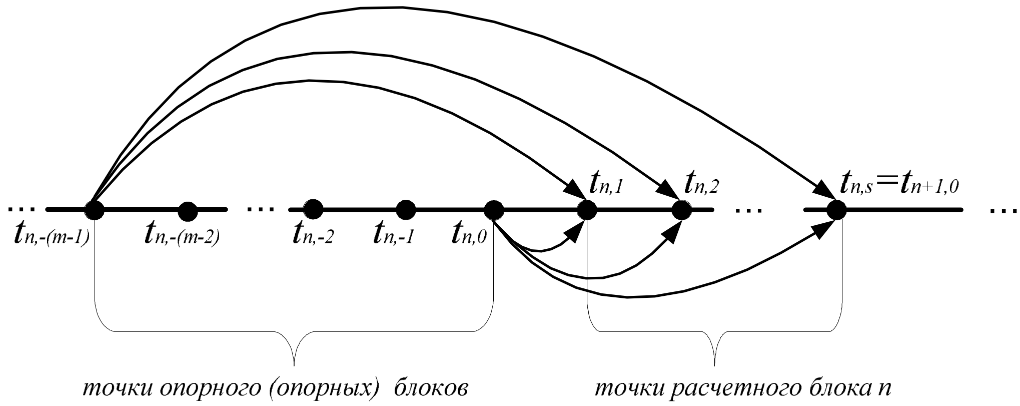


Рисунок 2 – Шаблон разностной схемы *m*-шагового *s*-точечного коллокационного блочного метода

Уравнения многошаговых разностных методов для блока из *s* точек при использовании вычисленных значений приближенного решения в *m* предшествующих блоку узлах, с учетом введенных выше обозначений, можно записать в виде:

$$\frac{1}{\tau}\sum_{j=1-m}^{s} a_{i,j}u_{n,j} = \sum_{j=1-m}^{s} b_{i,j} f_{n,j} = i = 1,2,...,s, \;\; n = 1,2,... \tag{2}$$

Для одношаговых методов, представляющих частный случай (2), если $m=1$, разностные уравнения имеют вид

$$\frac{1}{\tau}\sum_{j=0}^{s} a_{i,j}u_{n,j} = \sum_{j=0}^{s} b_{i,j} f_{n,j} = i = 1,2, \; ... \;, s, \quad n = 1,2, \; ... \tag{3}$$

Формулы (3) определяют одношаговый *s*-точечный разностный метод. Обозначим матрицы коэффициентов системы уравнений (2) через

$$A = \{ a_{i,j} \}, \quad B = \{ b_{i,j} \}, \; i = 1,2,...,s, \;\; j = -(m-1), -(m-2), ..., s.$$

Разобьем каждую из матриц на две части

$$A_1 = \{ a_{i,j} \}, \quad B_1 = \{ b_{i,j} \}, \; i = 1,2,...,s, \;\; j = -(m-1), -(m-2), ..., 0.$$

$$A_2 = \{ a_{i,j} \}, \quad B_2 = \{ b_{i,j} \}, \; i = 1,2,...,s, \;\; j = 1,2,...,s.$$

Введем соответствующие вектора

$$U_n = \{ u_{n,j} \}, \quad j = -(m-1), -(m-2), \ldots, 0,$$

$$V_{n+1} = \{ u_{n+1,j} \}, \quad n = 1,2,\ldots, \; j = 1,2,\ldots,s,$$

$$F_n = \{ f_{n,j} \}, \quad j = -(m-1), -(m-2), \ldots, 0,$$

$$F_{n+1} = \{ f_{n+1,j} \}, \quad n = 1,2,\ldots, \; j = 1,2,\ldots,s.$$

В векторной форме уравнение (2) будет иметь вид

$$A_1 U_n + A_2 V_{n+1} = \tau (B_1 F_n + B_2 F_{n+1}), \qquad n = 1,2, \ \ldots \tag{4}$$

Чтобы система однородных разностных уравнений, соответствующая (4), была линейно независима, потребуем, чтобы матрица A_2 была невырожденной, и разрешим ее относительно V_{n+1}

$$V_{n+1} = QU_n + \tau (DF_n + GF_{n+1}), \tag{5}$$

где $Q = -A_2^{-1} A_1, D = A_2^{-1} B_1, G = A_2^{-1} B_2$.

Задав стартовые значения, полученные, например, с помощью явных стадийных схем, необходимо на каждом последующем этапе решить нелинейное уравнение (5), определив последовательно вектора $V_1, V_2, \ldots$.

Представление блочных разностных многошаговых многоточечных уравнений в виде (5) будем называть канонической формой их записи. Получим коэффициенты уравнений, позволяющие представлять методы (2) в канонической форме (5). Каждое *i*-е разностное уравнение

$$u_{n,i} = \sum_{j=1-m}^{0} q_{i,j} u_{n,j} + \tau (\sum_{j=1-m}^{0} d_{i,j} f_{n,j} + \sum_{j=1}^{s} g_{i,j} f_{n,j}), \; i = 1,2,\ldots,s \tag{6}$$

содержит *2m+s* неизвестных коэффициентов $q_{i,j}, d_{n,j}, \; j = -(m-1), -(m-2), \ldots, 0$ $g_{n,j}, \; j = 1,2,\ldots,s$. Для их определения следует использовать *2m+s* уравнений условий аппроксимации. Выражения для невязок на решении *x(t)* исходного дифференциального уравнения имеют вид

$$r_{n,i} = \frac{1}{\tau}(-x_{n,i} + \sum_{j=1-m}^{0} q_{i,j} x_{n,j}) + \sum_{j=1-m}^{0} d_{i,j} x'_{n,j} + \sum_{j=1}^{s} g_{i,j} x'_{n,j}, \tag{7}$$

где $x_{n,i} = x(t_n + j\tau), \; x_{n-1,m} = x_{n,0}, \; x'_{n,i} = x'(t_n + j\tau) = f(t_n + j\tau),$ $x'_{n-1,m} = x'_{n,0}$.

Для каждого *i-го* уравнения потребуем его аппроксимации в точке $t_{n,0}$

$$x_{n,j} = x(t_{n,0} + j\tau), \quad x'_{n,j} = x'(t_{n,0} + j\tau) = f(t_{n,0} + j\tau, x_{n,j}).$$

Раскладывая $x_{n,j} = x(t_{n,0} \pm j\tau)$ и $x'_{n,j} = x'(t_{n,0} + j\tau)$ в ряды Тейлора в окрестности точки $t_{n,0}$, подставляя эти разложения в выражение (7), группируя члены с одинаковыми степенями по τ, а затем приравнивая нулю коэффициенты при τ^l, получим систему уравнений

$$\sum_{j=1-m}^{0} q_{i,j} = 1, \quad i = 1,2,...,s,$$

$$\sum_{j=1-m}^{0} (j\, q_{i,j} + d_{i,j}) + \sum_{j=1}^{s} g_{i,j} = i,$$

$$\sum_{j=1-m}^{0} (q_{i,j} \frac{j^l}{l} + g_{i,j}\, j^{l-1}) + \sum_{j=1}^{s} g_{i,j}\, j^{l-1} = \frac{i^l}{l},\ l = 2,3,...,p. \quad (8)$$

Поскольку каждое i-е разностное уравнение (6) содержит $2m+s$ неизвестных коэффициентов, то максимальный порядок аппроксимации рассматриваемого m-шагового s-точечного разностного метода равен $p=2m+s-1$.

Каноническая форма одношаговых многоточечных уравнений будет иметь соответственно вид

$$u_{n,i} = u_{n,0} + \tau(d_{i,0} f_{n,0} + \sum_{j=1}^{s} g_{i,j} f_{n,j}),\ i = 1,2,...,s. \quad (9)$$

Коэффициенты разностных уравнений (6) и (9) можно определить, решая систему (8) для общих многошаговых блочных методов, учитывая их частный вид, или интегро-интерполяционным методом. Для этого строится интерполяционный многочлен $L_{m+s-1}(t)$ с узлами интерполяции $t_{n,j-m}$ и соответствующими им значениями сеточной функции $u_{n,j}, j = -(m-1),-(m-2),...,0,1,...s$. Находятся производные полученного интерполяционного многочлена в узлах сетки $t_{n,i}, i = 1,2,...,s$. Приравняв их соответствующим значениям правой части дифференциального уравнения, получим разностные уравнения для блока. Используя изложенный выше подход, можно сформировать каноническую систему уравнений для любых значений параметров m и s. В частности, для параметров $m=3,\ s=2,\ p=7$ и шаблона, приведенного на рис. 3, система будет выглядеть следующим образом

$$\begin{pmatrix} v_{n,1} \\ v_{n,2} \end{pmatrix} = \tau \left(\begin{pmatrix} \frac{141 f_{n,-2}}{1360} + \frac{207 f_{n,-1}}{170} + \frac{297 f_{n,0}}{170} \\ -\frac{36 f_{n,-2}}{85} - \frac{336 f_{n,-1}}{85} - \frac{806 f_{n,0}}{85} \end{pmatrix} + \begin{pmatrix} \frac{57 f_{n,1}}{170} - \frac{9 f_{n,2}}{1360} \\ \frac{144 f_{n,1}}{85} + \frac{24 f_{n,2}}{85} \end{pmatrix} \right) +$$

$$+ \begin{pmatrix} \frac{55 u_{n,-2}}{136} + \frac{27 u_{n,-1}}{17} - \frac{135 u_{n,0}}{136} \\ -\frac{54 u_{n,-2}}{17} - \frac{128 u_{n,-1}}{17} + \frac{199 u_{n,0}}{17} \end{pmatrix}.$$

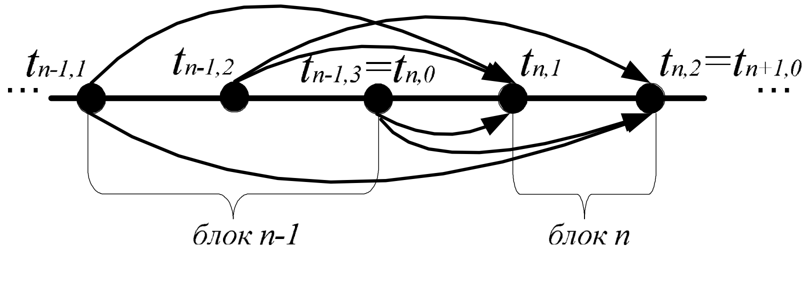


Рисунок 3 – Шаблон разностной схемы коллокационного 3-шагового 2-точечного блочного метода

Управление шагом при интегрировании блочными методами

Рассмотрим решение задачи (1) одношаговым коллокационным блочным методом (9) с числом расчетных точек *s*. Выделим две системы процессорных узлов, на которых запустим параллельные процессы. Первая система узлов будет обеспечивать параллельную реализацию одношагового коллокационного блочного *s*-точечного метода, что потребует *s* процессорных элементов для получения значений $u_{n+1,i}$, $i = 1,2,...,s.$ На второй системе узлов будет осуществляться реализация одношагового коллокационного блочного $s+1$-точечного метода с получением значений $\nu_{n+1,i}$, $i = 1,2,...,s+1.$ Поскольку методы являются неявными, каждый временной шаг подразумевает проведение некоторого количества итераций, обеспечивающих требуемую локальную точность.

Реализация обоих методов будет осуществляться автономно, что обеспечит крупнозернистость вычислений.

Размерность вычислительного поля может быть представлена двумя вариантами. В первом случае это могут быть кольцевые топологии, количество процессорных элементов в которых совпадает с размерностью блоков *s* и $s+1$ соответственно, во втором случае вычислительные поля представляют собой решетки процессорных элементов (рис. 4).

Первая решетка с *s* (количество точек в блоке) столбцами и *N* (количество уравнений в системе) строками, у второй решетки размерность процессорного поля – $(s+1)\times N$.

Если используется кольцевое вычислительное поле, размерность которого совпадает с размерностью блока, каждый процессорный элемент закрепляется за рассчитываемой точкой блока для всех уравнений системы.

Для осуществления итераций по (9) в каждом *i*-м процессорном элементе, обеспечивающем вычисления *s*-точечным методом, должны быть размещены соответствующие коэффициенты $a_{i,j}, b_i$, $i = 1,2,...,s,\ j = 1,2,...,s$, а также значения элементов вектора правых частей системы в последней точке предшествующего блока, которые будут считаться начальными для следующего.

Для группы процессорных элементов, которые обеспечивают решение $s+1$-точечным методом, соответствующие коэффициенты $a_{i,j}, b_i$, $i = 1,2,...,s+1,\ j = 1,2,...,s+1$.

В случае, если используется решетка процессорных элементов, позиция каждого элемента в строке определяет порядковый номер уравнения в системе, для которого ведется расчет, а номер элемента в столбце определяет рассчитываемую точку блока.

Поскольку вычисления проводятся для блоков точек, которые расположены регулярно, есть возможность сопоставления решений, полученных методами с разными порядками точности в *s*-совпадающих расчетных точках $t_{n+1}, t_{n+2},..., t_{n+s}$. Если норма вектора расхождений не превосходит заданную глобальную точность вычислений $\|u_{i,i} - \nu_{i,i}\| \le tol, i = 1,2,...,s$, за основу берется решение, полученное коллокационным блочным $s+1$-точечным методом $\nu_{n+1,i}$, $i = 1,2,...,s+1.$

Рассчитывается новое значение шага τ_{new}

$$\tau_{new} = \tau_{n+1}\, min\left(fax\,max, max\left(fax\,min, \omega\left(\frac{\varepsilon}{\left\| u_{i,i} - \nu_{i,i} \right\|} \right)^{\frac{1}{s+1}} \right) \right)$$

и по результатам расчета $s+1$-точечным методом формируется новый вектор опорных точек.

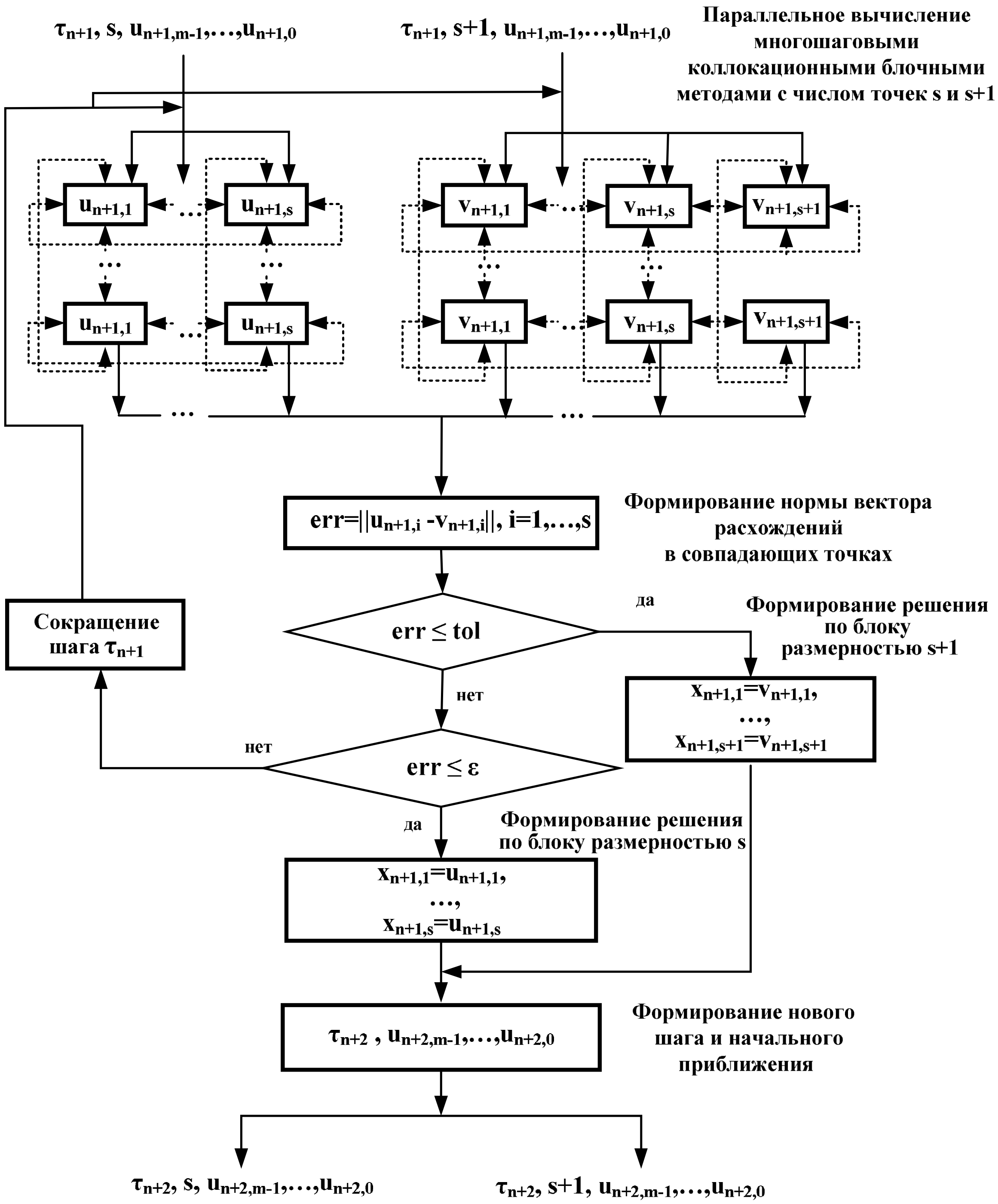


Рисунок 4 – Алгоритм управления шагом интегрирования на решетке процессорных элементов

Если норма вектора расхождений не превосходит заданную локальную точность вычислений, т.е. $tol < \left\| u_{i,i} - \nu_{i,i} \right\| \le \varepsilon, i = 1,2,...,s$, за основу берется решение,

полученное коллокационным блочным s-точечным методом, $u_{n+1,i}$, $i = 1,2,\ldots,s$. Рассчитывается новое значение шага τ_{new}, и новый вектор опорных точек формируется по результатам расчета s-точечным методом. Если полученное решение не обеспечивает заданную локальную точность, от шага необходимо отказаться, сократив его на величину, задаваемую параметром *faxmin*.

Параллельное управление шагом для задачи (1) многошаговым коллокационным блочным методом (6) с числом опорных точек m и расчетных s не будет иметь принципиальных различий с подходами, рассмотренными выше. Однако возникает необходимость в формировании начальных данных для расчета очередного блока значений. Для выполнения следующего шага нужно обеспечить доступ каждого процессорного элемента к значениям правых частей в опорных точках блока.

Кроме того, для осуществления итераций по (6) в каждом i-м процессорном элементе, обеспечивающем вычисления s-точечным m-шаговым методом, должны быть размещены соответствующие коэффициенты $a_{i,j}, b_{i,l}$, $i = 1,2,\ldots,s,\ j = 1,2,\ldots,s, l = 1,2,\ldots,m$, а также значения элементов векторов правых частей системы в m точках предшествующего блока, которые будут считаться начальными для следующего. Для группы процессорных элементов, которые формируют решение $s+1$-точечным методом, соответствующие коэффициенты $a_{i,j}, b_{i,l}$, $i = 1,2,\ldots,s+1,\ j = 1,2,\ldots,s+1, l = 1,2,\ldots,m$.

Поскольку максимальный порядок p аппроксимации рассматриваемого m-шагового s-точечного разностного метода, который равен $p=2m+s-1$, не обеспечивает абсолютной устойчивости метода [12], при генерации расчетных коэффициентов необходимо сократить предельное значение, введя параметр $q<p$. Тогда, в зависимости от значения нормы вектора расхождений, будет изменяться расчетная схема определения шага интегрирования и новое значение шага τ_{new} определяется с учетом поправки на порядок q аппроксимации.

Реализация тестовых задач на основе алгоритмов управления размером шага

Для экспериментов на мультиосновных машинах использовался программный интерфейс MathLink среды параллельных вычислений в Mathematica Wolfram Research, обеспечивающий возможность параллельного поиска решений как на локальной машине, так и в сетевом масштабе.

Также параллельная реализация осуществлялась на кластере NeClus MIMD-архитектуры с распределенной памятью.

Кластер состоит из 93-х вычислительных узлов, узла управления (Front Node), системы коммутации в составе двух гигабитных Ethernet-коммутаторов (Switch HP Procurve).

В качестве тестовой выбрана жесткая задача из [9]

$$x_1' = 2t * \sqrt[5]{x_2} * x_4,\ \ x_2' = 10t * e^{5(x_3-1)} * x_4, \qquad (10)$$

$$x_3' = 2t * x_4,\ \ x_4' = -2t * \ln x_4$$

с начальными условиями

$$x_1(0) = x_2(0) = x_3(0) = x_4(0) = 1$$

и с известными точными решениями

$$x_1(t) = e^{\sin(t^2)},\ x_2(t) = e^{5\sin(t^2)},\ \ x_3(t) = \sin(t^2) + 1,\ x_4(t) = \cos(t^2).$$

Поиск решения проводился на интервале [0, 2.5]. Из-за разных масштабов решения, особенно в середине интервала интегрирования, потребовалось использовать процедуру управления шагом, основанную на рассмотренных коллокационных блочных методах (рис. 5а). Рассматривались реализации на основе одношаговых коллокационных блочных методов с числом расчетных точек 2 и 3, и многошаговых – с числом опорных точек 2 и расчетных 2 и 3. Шаги осуществлялись параллельно, продвижение вдоль оси времени проходило сразу на размерность блока, обеспечивающего заданную точность, к середине интервала интегрирования заметно сокращение шага.

Вторая тестовая задача из [14]

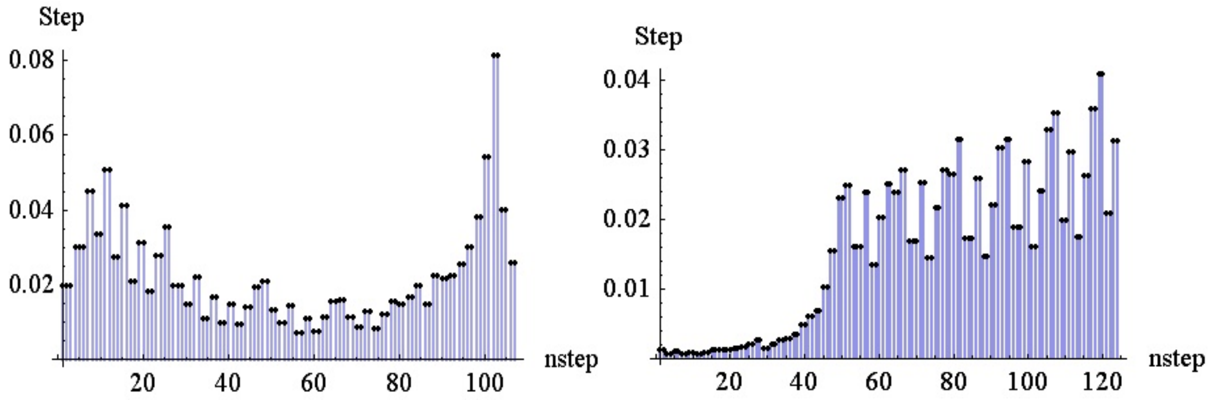
$$x_1' = -x_1 + x_2, \;\; x_2' = -100x_1 - x_2, \qquad (11)$$

$$x_3' = -100x_3 + x_4, \;\; x_4' = -10000x_3 - 100x_4*,$$

с начальными условиями

$$x_1(0) = 1, \quad x_2(0) = 0, \quad x_3(0) = 1, \quad x_4(0) = 0,$$

имеет проблемы при численном интегрировании в начале отрезка (рис. 5б)



а) б)

Рисунок 5 – Управление шагом интегрирования для тестовых задач (10) и (11)

Тестовые жесткие задачи третьего типа [15] возникают при дискретизации уравнения диффузии методом прямых. Количество уравнений в системе N может быть сколь угодно велико и определяется размерностью расчетной сетки, которая вводится для решения исходного уравнения в частных производных. Система (1) решалась на интервале [0, 4], правые части системы имеют вид

$$f(t,(x(t)) = (N+1)^2 \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 \end{bmatrix} x + (N+1)^2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ 0 \\ \varphi(t) \end{bmatrix}, \qquad (12)$$

где

$$\varphi(t) = \alpha\, sin\left(\sqrt{2}\right) exp(-\vartheta t) - sin(1) exp(-\mu t),$$

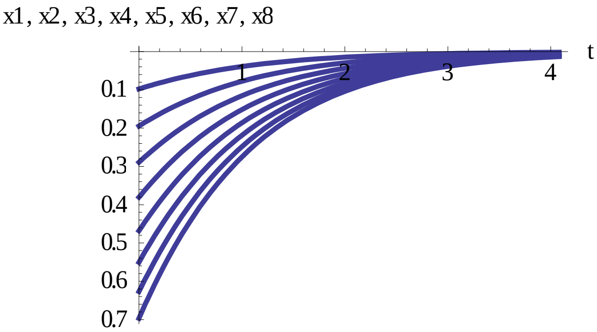
$$\alpha = cos\left(\sqrt{2}\right) / \sqrt{2}\, cos\left(\frac{1}{\sqrt{2}}\right),$$

$$\vartheta = 2(N+1)^2\left(1-\cos\left(\frac{\sqrt{2}}{N+1}\right)\right), \quad \mu = 2(N+1)^2\left(1-\cos\left(\frac{1}{N+1}\right)\right).$$

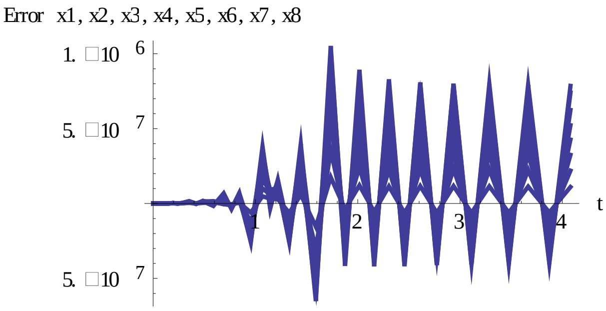
Точные решения

$$x_i(t) = \alpha\,\sin\left(\frac{\sqrt{2}}{N+1}\right)\exp(-\vartheta t) - \sin\left(\frac{1}{N+1}\right)\exp(-\mu t), \quad i = 1,2,\ldots,N\,.$$

На графиках приведены численные результаты решения (рис. 6а) и погрешности (рис. 6б), полученные для системы с N = 8, а также диаграмма параллельного управления шагом (рис. 7).



а)



б)

Рисунок 6 – Численное решение и погрешности системы (12)

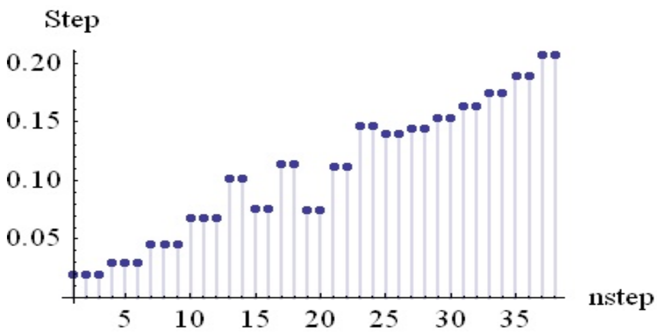


Рисунок 7 – Управление шагом интегрирования для тестовой задачи (12)

Выводы

В статье рассмотрены вопросы разработки коллокационных блочных методов решения задачи Коши, ориентированных на использование в параллельных вычислительных системах с распределенной памятью. Основная идея, на которой базировалось конструирование блочных методов для решения СОДУ на параллельных компьютерах, заключалась в одновременном получении приближений точного решения в точках, образующих блок. Параллельный счет осуществлялся в пределах одного цикла для всех точек блоков с размерностями s и $s+1$. Две нити вычислений проходили независимо, и необходимость в обменах возникала только после получения конечных результатов для блоков расчетных точек. Структура предлагаемых методов позволила построить алгоритмы автоматического управления шагом интегрирования, что является особенно актуальным при моделировании объектов, которые описываются жесткими или плохо обусловленными системами уравнений. Поскольку точки внутри блока расположены регулярно, стало возможным определение и сопоставление локальных

погрешностей во всех точках блока, в отличие от стадийных методов, где стадии, как правило, не совпадают, и сравнение ведется только по одной конечной расчетной точке.

На основе предложенных алгоритмов управления размером шага реализованы тестовые задачи, численное решение которых обеспечивает заданную точность с максимально возможным при этом шагом интегрирования.

Для экспериментов на мультиосновных машинах использовался программный интерфейс MathLink среды параллельных вычислений в Mathematica Wolfram Research, обеспечивающий возможность параллельного поиска решений как на локальной машине, так и в сетевом масштабе.

Литература

1. Firsova A. Dynamic System Simulation. Robust algorithms of state estimation of dynamic lumped parameters systems / A. Firsova, O. Dmitrieva. – Lambert Academic Publishing, 2011. – 92 p.
2. Zanariah A.M. Solving Large Systems of Ordinary Differential Equations on Parallel Computer/ A.M. Zanariah, M.B. Suleiman // Journ. of Scientific Research. –2009. – Vol. 29, № 4. – P. 491-501.
3. Дмитриева О.А. Параллельное моделирование жестких систем на основе диагонализации полной матрицы / О.А. Дмитриева // Искусственный интеллект. – 2011. – № 4. – С. 46-53.
4. Feldman L.P. Embedded block parallel methods for initial Cauchy problem numerical solution / L.P. Feldman, I.A. Nazarova, O.A. Dmitrieva [et al.] // Proceedings of DNTU. – 2010. – № 1. –P. 12-17.
5. Soderlind G. Digital filters in adaptive time-stepping / G. Soderlind // ACM Trans. Math. Softwar. – 2003. – Vol. 29. – P. 1-26.
6. Дмитриева О.А Параллельное моделирование динамических объектов с автоматическим выбором шага на основе экстраполяционных методов / О.А. Дмитриева // Радіоелектронні і комп'ютерні системи. – 2012. – № 6 (58). – С. 312-317.
7. Дмитриева О.А. Параллельное моделирование жестких динамических систем диагонально-неявными методами с адаптацией шага / О.А. Дмитриева, Я.А. Куприй // Наукові праці ДонНТУ. ІКОТ-2010. Випуск 12(165). – Донецьк : ДонНТУ. – 2010. – С. 111-116.
8. Дмитриева О.А. Управление шагом в блочных диагонально-неявных методах решения обыкновенных дифференциальных уравнений / О.А. Дмитриева // Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика та обчислювальна техніка» (ІКОТ-2010). Випуск 11(164). – Донецьк : ДонНТУ. – 2010. – С.14-18.
9. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Жесткие задачи / Э. Хайрер, Г. Ваннер. - М. : Мир, 1999. – 685 с.
10. Burrage K. Parallel iterated method based on Variable stepsize Multistep Runge-Kutta methods of Radau type for stiff problems / K. Burrage, H. Suhartanto // Adv. Comput. Math. – 2000. – Vol. 13. – P. 257-270.
11. Дмитриева О.А. Упрощение итераций при параллельной реализации неявных методов решения систем обыкновенных дифференциальных уравнений / О.А. Дмитриева // Наукові праці Донецького національного технічного університету. Серія «"Інформатика, кібернетика та обчислювальна техніка» (ІКОТ-2011). Випуск 13(185). – Донецьк : ДонНТУ. – 2011. – С. 13-18.
12. Дмитриева О.А. Разработка обобщенных коллокационных блочных методов / О.А. Дмитриева // Сборник трудов конференции МОДЕЛИРОВАНИЕ – 2012. 16 -1 8 мая 2012, г. Киев. – Киев : Институт проблем моделирования в энергетике, 2012. – С. 434-437.
13. Дмитрієва О.А. Генерація стійких блокових методів для розв'язання жорстких диференційних рівнянь і їх систем / О.А. Дмитрієва // Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика та обчислювальна техніка» (ІКОТ-2011). Випуск 14(188). – Донецьк : ДонНТУ. – 2011. – С. 36-43.
14. Enright W.H. Software for ordinary and delay differential equations: Accurate discrete approximate solutions are not enough / W.H. Enright //Applied Numerical Mathematics. – 2006. – Vol. 56, № 3-4. – P. 459-471.
15. Butcher J.C. ARK methods for stiff problems / J.C. Butcher, C.N. Rattenbury // Applied Numerical Mathematics. – 2005. – Vol. 53, № 1. – P. 165-181.

Literatura

1. Firsova A, Dmitrieva O. Dynamic System Simulation. Robust algorithms of state estimation of dynamic lumped parameters systems. – Lambert Academic Publishing. 2011. 92 p.

2. Zanariah A. M., Journ. of Scientific Research. 2009. Vol. 29. № 4. P. 491-501.
3. Dmitrieva O. A. Iskusstvennyj intellect. 2011. № 4. S. 46-53.
4. Feldman L., Nazarova I., Dmitrieva O., Mikhaylova T. Proceedings of DNTU. 2010. №1. P.12-17.
5. Soderlind G. ACM Trans. Math. Softwar. 2003. Vol. 29. P. 1-26.
6. Dmitrieva O. A. Radioelektronnye komp'yuternye systemy. 2012. № 6 (58). S. 312-317.
7. Dmitrieva O. A., Kupriy J.A. IKOT. 2010. №12 (165). S. 111-116.
8. Dmitrieva, O. A. IKOT-2010. № 11 (164). S.14-18.
9. Hayrer E. Wanner. G. Solution of ordinary differential equations. Tough task. - Springer-Verlag, 1999.685s.
10. Burrage, K. Adv. Comput. Math. 2000. Vol. 13. P. 257-270.
11. Dmitrieva O. A. IKOT-2011. № 13 (185). S. 13-18.
12. Dmitrieva O.A. Sbornik trudov conferencii MODELIROVANIE. 2012. 16-18 May. Kiev. S.434-437.
13. Dmitrieva O.A. IKOT-2011. № 14 (188). S. 36-43.
14. Enright W.H. Applied Numerical Mathematics. 2006. Vol. 56. № 3-4. P. 459-471.
15. Butcher J. C. Applied Numerical Mathematics. 2005. Vol. 53. № 1. P. 165-181..

RESUME

O.A. Dmitrieva

High-Performance Algorithms for Step Control Based on the Parallel Collocation Block Methods

The paper discussed the issue of the design of collocation methods for solving Cauchy problem, which are oriented for using in parallel computers with distributed memory. The main idea, which underlines the development of the block methods for solving SODE on parallel computers, was to obtain simultaneously the approximation of the exact solution at points forming the block. A parallel count was performed within one cycle for all points of blocks with dimensions of s and $s+1$. Two flows of computation were performed independently, and the need for exchanges occurred only after obtaining the final results for blocks of the estimated points. The structure of the proposed methods allowed us to design the algorithms for automatic control of integration step, which is particularly relevant for modeling objects, which are described by hard or ill-conditioned systems of equations. Since the points are located regularly inside the block, it became possible to identify and compare local errors at all points of the block, in contrast to the stage methods, where the stages are usually not the same, and the comparison is made only by one final design point.

On the basis of the proposed algorithms of step size control the test problems were implemented, the numerical solution of which provides the required accuracy with the maximum possible integration step.

For the experiments on multi-based machines, MathLink programming interface of parallel computing environment in Mathematica Wolfram Research was used, which provides the possibility of parallel search for solutions on the local machine as well as in the network scale.

Статья поступила в редакцию 05.06.2012.